Int J Soc Robot (2009) 1: 233–242
DOI 10.1007/s12369-009-0023-5

ORIGINAL PAPER

Combining Moral Theory, Modal Logic and Mas to Create Well-Behaving Artificial Agents

Vincent Wiegel · Jan van den Berg

Accepted: 20 May 2009 / Published online: 5 June 2009

Abstract Witnessing a growing number of increasingly autonomous software agents we interact with or that operate on our behalf under circumstances that are not fully known in advance, we argue that there is a need to provide these agents with moral reasoning capabilities. Looking at the current literature on behaviour constraints and multi-agent (software) systems (MAS), one can distinguish various topics. The first topic concerns the analysis of various forms of restraint and their basis. This topic is at the core of moral philosophy. The second topic concerns the formalized specification of, and the reasoning about the constraints. The research on this topic focuses predominantly on the use of logic, mostly modal logic, and defeasible logic. The last topic is the MAS and implementation related topic of designing a working system in which there are rules that can be enforced and deviant behaviour be detected.

Here we argue that all three topics need addressing and strong integration. The moral philosophical analysis is needed to provide a detailed conceptualization of the various forms of behaviour constraint and direction. This analysis goes beyond what is usual in the more technical/design focus. The (modal) logic provides the rigour required to ultimately allow implementation. The implementation itself is the ultimate objective. We outline the three components and demonstrate how they can be integrated. We observe here that we do not intend, or claim, that this moral reasoning is on par with human moral reasoning. Our claim is that the analysis of human moral reasoning may provide a useful model for constraining software agent behaviour. And, as equally important, it is recognizable by humans which is an important characteristic when it comes to 'human–artificial agent' interaction. Recognizing and understanding the precise basis for the behaviour constraint in the artificial entity will make the agent more trustful which, in its turn, will facilitate the acceptance of the use of and the interaction with artificial agents.

Keywords Software agents · Morality · Modal logic · Trust · Deontic constraints · BDI

V. Wiegel · J. van den Berg (✉)
Delft University of Technology, Faculty of Technology, Policy and Management, Delft, The Netherlands
e-mail: j.vandenberg@tudelft.nl

V. Wiegel
e-mail: v.wiegel@tudelft.nl

1 Introduction

Thinking about Human–Robot Interaction, quite a lot of artificial intelligent researchers have wished to build robots or softbots that seem to think, feel and even live, and that it would become possible that friends of these creatures can share some of their lives with them [19]. We use the term artificial agent to refer to both robots and softbots (or software agents). In our paper we discuss software constructions since morally reasoning in our current scope has no material aspects except in the trivial sense of requiring computer hardware to run on.

Looking at most practical research efforts however, this optimistic picture turned out to be hard to achieve and researchers often confined themselves to design agents with just very limited and strongly context-dependent capabilities like playing (just one type of) a game (like playing chess [17]), searching a cheap air plane ticket or finding information on a subject of your choice. We admit that, fortunately,

more sophisticated research efforts are also available including on (adaptive) negotiation agents for performing transaction in e-business [19] and agents that monitor all kinds of (complex) systems like inventory systems, planning and scheduling systems, and large computer networks [27].

When trying to design more complex human-like agents, one rapidly encounters all kinds of additional issues that should be dealt with including trust, emotion, adaptability, agility, resilience and other both basic and fundamental capabilities [3, 7, 8]. Especially in circumstances with a lot of vagueness and uncertainty where we admit autonomous decision making by agents (e.g., automatic landing of an airplane under very difficult weather conditions or the decision whether or not to close a storm surge barrier [21]), difficult trade-offs should be made by these agents which may include difficult ethical considerations.

With respect to robot–human relationships trust is a key concept. There are various modes in which humans and artificial agents can interact. First, humans can delegate actions to artificial agents, for example, buying goods, finding information, protecting property. Second, humans process the results from the delegated actions by evaluating them, accepting or rejecting them, etc. Third, there is a form of companionship in which human and artificial agents play a game, talk and listen, or just share a physical space. This is only a sketchy outline of what is possible. The overview serves to illustrate that the notion of trust is present in each of these modes of interaction. Looking at what it means to trust and to be trustworthy there are several elements that are at the core.[1] Trust is an act or an attitude of the one who is trusting someone or something else. Trustworthiness is a property of the one, or the thing that is being trusted, the trustee. Trusting something else means accepting a risk of not getting what one was expecting the trustee to accomplish. This failure is not just due to an ability to deliver but also to some form of betrayal. When a machine that is used to accomplish something does not work one can feel disappointed (one *relied* on the machine) but not betrayed. Betrayal needs not be limited to human–human relationships, but *can* also apply to human–artificial agent relationships, that is transcend beyond a simple use relationship. Another element in trust is the expectancy of competency. For trust to be warranted there must be a reasonable expectation that the trustee is capable of executing the task that it is being delegated. The expectation might be in part based on the information provided regarding earlier performances. If this information is misleading there is again reason to feel betrayed. Yet another element of trust, and a more controversial one, is the intention of the trustee to act in accordance with the expectations and to the good of the truster. This might not be necessary. A truster that is motivated by a contract or self-interest can properly execute the delegated task. On all accounts the deontic constraints of the trustee are important in gaining the attribute of trustworthiness. Knowing that it is under an obligation to preserve privacy, for example, might restrain an artificial agent in disseminating information that the truster provided.

The importance of the trust and morality motivated us to start a research program that includes the investigation of autonomous agents with moral reasoning and decision making capabilities. In this paper, the focus is on morally motivated behaviour constraint in artificial agents that are acting in (quasi)-autonomous ways in all kinds of different situations, in which their precise behaviour is usually not completely known at design time. Their actions are chosen at run-time under circumstances that could have been foreseen, i.e., not have been analysed explicitly beforehand when they were designed. Therefore, their behaviour might have undesirable (possibly unintended) consequences. The consequences might even be malicious. One way to partially address this issue is by extending agents with a moral reasoning capability. This may be realized in the form of an internalised behaviour constraint (as we will elaborate below) as opposed to a type of externalised behaviour constraint in, for example, the use of reputation or behaviour enforcement.

Looking at the current literature on behaviour constraints and multi-agent (software) systems (MAS), one can distinguish various topics. The first topic concerns the analysis of various forms of restraint and their basis. This topic is at the core of moral philosophy (see for example [16]). The second topic concerns the formalized specification of, and the reasoning about the constraints. The research on this topic focuses predominantly on the use of logic, mostly modal logic (see for example [6, 9–11, 20]), and defeasible logic (see for example [12, 14]). The last topic is the MAS and implementation related topic of designing a working system in which there are rules that can be enforced and deviant behaviour that can be detected (see for example [2]).

In several publications one will find in-depth discussions on one of these topics, with some tentative links to the other ones. For example, Dastani et al. put strong emphasis on the modal logic modelling of agent behaviour. The research has some links to implementation questions, but that link is rather weak as is the link to moral philosophy. Artakis, on the other hand, is very strong on MAS and the implementation component. Its logical component is less extensive, while the tie to moral philosophy is nonexistent. Wooldridge [25, 26] is very strong on the logical aspect while the MAS focus is more on the conceptual level

[1] Here also our aim is not to provide an analysis of the notion of trust. We merely use it to demonstrate our approach. There are many different views. Our presentation is not meant as a position in a debate on what trust is or involves. The presentation is informed by McLeod, Carolyn, "Trust", The Stanford Encyclopedia of Philosophy (Winter 2006 Edition), Edward N. Zalta (ed.), URL: http://plato.stanford.edu/archives/win2006/entries/trust/.

than on the actual implementation. There is no conceptual–philosophical analysis to underpin the behaviour regulation. The same goes for Georgeff [13].[2]

In this paper we argue that all three topics need addressing and strong integration. The moral philosophical analysis is needed to provide a detailed conceptualization of the various forms of behaviour constraint and direction. This analysis goes beyond what is usual in the more technical/design focus. The (modal) logic provides the rigour required to ultimately allow implementation. The implementation itself is the ultimate objective. We outline the three components and demonstrate how they can be integrated. We observe here that we do not intend, or claim, that this moral reasoning is on par with human moral reasoning, nor that it should be like human moral reasoning. Our claim is at most that the analysis of human moral reasoning may provide a useful model for constraining software agent behaviour. And, it is recognizable by humans, which is an important characteristic when it comes to 'human agent-software agent' interaction.

Bringing the above-mentioned three topics together and integrating them into one coherent approach is the goal of this paper. In addition, we show how this approach can be used to implement autonomous agents that have internalized constraints on their behaviour in order to guard against undesirable behaviour. During implementation we also look at normative moral reasoning from a meta-level position. With respect to morality we will disregard questions about both its source (human, divine, etc.), its nature (natural property, emergent characteristics, etc.) and its particular object (human, animal, artificial, etc.). In our approach we focus on moral knowledge and action, and their structure. We relate the knowledge about what is morally (un)acceptable to the desires (goals) of the agents and the formation of intentions (adaptation of plans) by agents. In order to do this we use the belief-desire-intention model (BDI-model) by Bratman [4, 5], and an extensive framework of modal logic, DEAL. In this paper we only provide an informal description and not a complete formalization as the main goal is just to demonstrate the integral approach. For testing purposes some implementations are done using a particular multi-agent software system named JACK [1]. In the paper we represent some of this effort using pseudo-code.

The remainder of this paper is structured as follows: In section two, we sketch our integral approach. Section three details the approach and focuses on the implementation using an example. Section four concludes the paper with a first evaluation and outlook.

[2]We are fully aware that the above outline is a gross simplification and far from complete. We hope for the current purposes it suffices to make our point of the three components and the need for integration.

2 The Integral Approach

2.1 Moral Philosophical Considerations

To implement autonomous agents with moral reasoning capabilities based on internalized behaviour constraints, we first need to choose an appropriate type of moral philosophy. We first provide a rough characterization of moral philosophy and some aspects that are pertinent when considering implementation.

Moral philosophy has several branches, amongst which meta-ethics, applied ethics and normative ethics. Meta-ethics deals, amongst other things, with the possibility of ethics, what moral knowledge is, and how it is possible, if at all, to have moral knowledge. In our discussion, however, we just use ethics as a suitable model without having to engage in the meta-ethical discussions on the possibility of it because our targets are not human but artificial agents. Therefore, meta-ethics is considered less relevant for our purposes.

Applied ethics focuses on specific domains, such as medical applications, environmental issues, etc. When it comes to concrete MAS applications in specific domains, applied ethics will be a useful field to draw upon. Here, in the context of this paper however, the specific issues of applied ethics do not play an important role yet. Hence, we will for now leave applied ethics out of scope.

For our current general discussion, normative ethics (which, roughly speaking, deals with what is to be considered as right and wrong conduct) provides us all the input we need. Normative ethics itself can be divided into three broad categories: teleological ethics, virtue ethics, and deontological ethics. Teleological ethics looks at the consequences of an act to evaluate it morally: 'the outcome justifies the action'. E.g., utilitarianism is a well-known form of teleological ethics.

Virtue ethics looks at the character traits of an individual for its moral evaluation. Wisdom, integrity and bravery are well-known operators in virtue ethics. However, since artificial agents are usually modelled in terms of concrete actions they perform, virtue ethics seems to be farfetched when discussing and evaluating the moral outcomes of their behaviour. It is also doubtful whether the current technology allows us to construct virtuous artificial agents.

In deontological ethics the focus is on the action or state in its own right ('killing is wrong' and 'lying is wrong', no matter what the consequences are). Kant is the best-known representative of the approach. To understand why deontological ethics does offer a suitable tool for modelling moral reasoning of artificial agents, we consider moral considerations under two roles that are relevant in the context of behaviour direction. First, moral goals can be considered on par with non-moral goals. 'To do good', such as helping the

poor, can be a goal in itself. As such, they are taken into consideration with other goals and, when using the same resource, subject to choice. Second, moral considerations can also act as a constraint on actions. This concerns desiring something but not doing it because the action itself, or some aspects of the resulting state, are thought to be morally undesirable.

In our current context there are two differences between teleological and deontological ethics that are very important. First, teleological ethics requires an estimation of the outcome of an action. In particular in complex situations this adds a considerable cognitive burden which is much less present for deontological considerations. Second, deontological theories allow more easily the exclusion of particular categories of actions. Since artificial agents (in the near future) are most likely to be special purpose agents whose primary focus is not on 'doing good' per se as moral benefactors but more on doing certain actions that do not violate moral rules. The focus is on morally acceptable actions that help us rather than explicitly improving the moral standard of the places we live in. Our choice to start with deontological ethics is thus based on pragmatic considerations rather than moral philosophical ones.

Another aspect to observe is that an act or outcome is never, or not completely, an act or outcome in its own right or per se. The physical, biological act of uttering some sounds has no moral bearings. It is only in the context of two members of a, say an English speaking community, that the utterance of sounds with a particular meaning becomes an offensive, morally deplorable act. A moral outcome of a given action is a particular aspect of a given situation that is morally right or wrong. It is an attribute or predicate we attach, or that is attached, or that is a property of the state. So if you are yelled at unprovoked, you and anyone else around will without any problem recognize this as morally undesirable. The reason for raising this point is that for an artificial agent this is far from trivial. Its sensors will detect sounds and images without further moral qualification. And it will perform acts without any moral consideration. Hence, everything it does or senses will somehow have to be given a moral weighing. The cognitive burden that is thus introduced in the construction of well/behaved artificial agents can hardly be underestimated.

The next aspect that is relevant is the distinction that can be made between actions and moral judgements that are immediate, and those that are meditated. By meditated we mean that some explicit consideration process precedes the actual decision to act or morally evaluate, whereas the instinctive action is chosen and executed without further consideration. For example, anyone witnessing the torture of an animal will repulse and have an immediate moral evaluation of the act. Whether to treat someone for an injury that resulted from a reckless action and for which the person in question cannot pay might require some explicit consideration. A moral artificial agent will have to have various modes of morally guided responses to allow for proper responses in different situations. Of course, the particular application will often help determine whether both modes are required or not.

Above we have given some informal characterizations of concepts that play an important role in moral reasoning. Concepts, moreover, that we believe help constructing an approach to constrain the behaviour of autonomous agents. Next, we discuss how these concepts can be formalized, a necessary condition for subsequent implementation of moral concepts.

2.2 Modelling

The first step towards implementation is the modelling of the required behaviour. For this purpose, DEAL (deontic epistemic action logic) is used in conjunction with the BDI-model. These models consist of standard modal logic operators. They are used as specification language. This provides a language to capture requirements that are stricter than our everyday language, but more relaxed than the logic reasoning with axiomatizing and theorem proving. Here we provide a rudimentary overview only. For a more detailed discussion the reader is referred to Van den Hoven and Lokhorst [16], Wiegel [23, 24], Halpern [15], Moor [18].

Reasoning about what one knows or believes is captured by epistemic logic, which has two operators: Bi (agent i believes that) and Ki (agents i knows that). $Ki(\Phi)$ states that agent i knows that Φ. Action logic has as its operator $STIT$, 'see to it that'. For example, $[iSTIT\colon \Phi]$ means that agent i sees to it that Φ is done or brought about.

Deontic logic is the logic of obligations. The obligation operator, $O(\Phi)$ it is obligatory that Φ, is often taken as the primitive operator. Others can be derived from this operator. $P(\Phi)$, it is permissible that Φ, or alternatively $\neg O(\neg\Phi)$, and $F(\Phi)$, it is forbidden that Φ, or alternatively $O(\neg\Phi)$, (Van den Hoven and Lokhorst, [17, 284]). Further distinctions are gratuitous actions G, $\neg O(\Phi)$ and optional actions Opt, $(\neg O(\Phi) \wedge \neg O(\neg\Phi))$.

For the formalization of the BDI-model, we follow Wooldridge's definitions [25]; $(Bel\ i\ \Phi)$ means i believes Φ, $(Int\ i\ \Phi)$ means i intends Φ, $(Des\ i\ \Phi)$ means i desires Φ. In addition to the modal logic operators, we use standard first-order logic. This overlaps with the epistemic component of DEAL.

All these elements can be combined to construct moral propositions. Consider the following proposition, which is for demonstration purposes only and not necessarily true or a desirable property of an artificial agent:

$$Bi(G(\Phi)) \rightarrow O([i\ STIT\ \Phi]) \tag{1}$$

This proposition means that if i believes that Φ is morally good, then i should act in such as way that Φ is brought about.

2.3 Implementation

To create support for the above modelling components we use the following implementation elements from the JACK development environment [1]:

- Beliefsets—beliefs representing the epistemic dimension
- Events—goals and desires, for the goal-directed behaviour
- Actions, plans and reasoning methods—representing the intentions and action logic
- Agents—the container for the other elements
- Java programming language.

Beliefsets can be modelled using 'open world' and 'closed world' semantics. In closed world semantics something is either true or false. The open world semantics allow something to be unknown. In the implementation the closed world beliefsets contain only tuples that are true. Tuples that are not stored are assumed false. In open world semantics both true and false tuples are stored. Tuples not stored are assumed unknown.

Desires, as represented by BDIGoalEvents in JACK, are a special type of events. Events can be inter-agent or intra-agent. The former represent the usual interaction between entities, the exchange of information, requests and answers. The latter represents fine-grained internal reasoning processes.

An agent has one or more plans at its disposal to achieve its goals. A plan is a sequence of atomic acts that an agent can take in response to an event. Committing to a plan, choosing a plan is like forming an intention. There are potentially several plans that can handle an event, and each plan can handle only one type of event. In order to determine which plan will handle an event (if any) there are two methods: *relevance*() and *context*(). The *relevance*() method determines which instances (all or some) of an event type can be handled. An event can carry various information which allows the *relevance*() method to determine whether or not to handle the event. From all relevant plans, the *context*() method determines next which are applicable. The context method is a logical expression that tries to bind the plan logical members. For each binding a plan instance will be created.

A plan can have some meta information associated to it—accessible through *PlanInstanceInfo*(). This can be a ranking number that can be given a cardinal or ordinal interpretation. This information can be used to reason at a meta-level in case there are multiple, applicable plans.

Having presented the basic elements needed for implementing autonomous agents with certain moral reasoning capabilities, we finally propose a methodology consisting of four steps in order to achieve a successful implementation. The four steps are:

(1) modelling of moral knowledge;
(2) building up a moral knowledge base which includes the mechanism for classifying actions and states under a particular moral view;
(3) integrating these components into the non-moral actions and states;
(4) adding meta-level, moral reasoning to make trade-offs between exclusive actions or outcomes.

In the next section, we illustrate the use of the integral framework on the basis of an application in the medical domain.

3 An Application

3.1 The Case

Consider the following, simplified, example. A hospital deploys artificial agents as medical data assistant to the physicians. The medical data assistant's duties comprise liaising with the patient, the patient's GP and the insurer. Assume further that the patient is not well insured and is poor, with a large family to maintain. Providing all patient data might allow the insurer to establish that the patient is not covered for a particular operation, in which case the patient cannot pay for it and the chances of the hospital having to forfeit payment increase. In this simple example we have to consider only one action: 'providing information'. As we argued above, an action is never completely a moral action in its own right but is context-dependent. The same action of providing information in this case can—depending on the precise context of who is informing who—be obligatory and forbidden, e.g., when it concerns patient data provided to the GP and to the insurer respectively. Furthermore we observe that an obligation to tell the truth does not automatically imply an obligation to inform (the insurer, for example). So this one single action of 'providing information' has many different moral aspects: duty to protect privacy, an obligation to inform the GP, a prohibition to inform the insurer, a prohibition to lie.

A choice the designer has to make is how to model these considerations under an appropriate normative ethical view. A teleological agent might decide that overall, the situation of the patient's family, the hospital's financial situation, etc. makes it permissible to withhold certain information. In the same way, in other situations the agent might decide to not uphold the privacy protection because that serves the overall benefit optimisation.

Under a deontological approach, however, these complicated considerations can be left out. Providing private patient data to the insurer is forbidden, providing the insurer with accurate data on the operation is obligatory, etcetera. These relatively simple considerations help to constrain the possible actions of information provision in a hospital environment. The overall set of obligations, permissions, with their conditions, the sub/sets of data they refer to, in constellations of many actors with many roles is complex already.

We argue that this latter, deontological approach will prove hard enough but doable. In gaining user acceptance of the use of robots and softbots in relation to humans it seems wise to err on the cautious side and build a robust approach that gains trust by not abusing information provided and applied through robots and softbots. Once proven reliable it can be extended with various other modes of moral consideration.

The importance of a robust approach can further be illustrated by extending the example. Consider the situation in which paramedical staff gets involved. The designer might not have foreseen it, or his sponsor might not have deemed it necessary to consider them at the time. Now suddenly, the medical data assistant has to consider the paramedics that are assisting our patient, who has had a car accident, and is being given first-aid on the spot. Or, consider the situation in which the patient was insured with a direct writer, but has recently switched to a broker mediated insurer. What is the status of the broker?

In this section we consider the ways to approach this application, and outline the impact of the various choices that can be made. Of course, as this is not the actual design, our outline will be necessarily incomplete and rough.

3.2 Modelling Moral Knowledge

Moral knowledge is modelled as an n-tuple beliefset with a series of attributes. The main of these are:

(1) the logical proposition;
(2) the validity domain, called the sphere loosely following Walzer's [22] use of the term sphere;
(3) a type indicator allowing the distinction between moral orientation (e.g. teleological or deontological orientation);
(4) the type of the object of the logical proposition will help classifying the proposition;
(5) a truth indicator.

```
1. Listing 'beliefset MoralObligations'
public beliefset MoralObligations extends
OpenWorld {
  #key field String strObligationName
  #key field String strSphere
  #value field String strMoralProposition
  #value field String strType
....
}
```

The moral knowledge itself can be formulated as follows. To model the obligation to inform the GP on a change in the state of the health of the patient, we define Δ as a change in the state of the patient health, $STIT\alpha$ as an action α to inform, and the state A as $A \stackrel{\text{def}}{=} Kg\Delta$ meaning that the state A describes that g knows Δ, where g is a group of agents or humans. Then the obligation to inform can be described as follows:

$$O(Bi(\Delta) \rightarrow ([i\ STIT\alpha\colon A])). \tag{2}$$

Statement (2) can be read as follows: If agent i believes the change of state Δ, then it has an obligation to act, namely to inform such that state A (here describing that group g knows Δ) is established through action α. Note that the action, to inform, is defined here in terms of the resulting state that g knows Δ. In the above formalization, the action is obligatory and not the state. The group g would be defined in terms of the relationship with the patient by using an appropriate predicate logic definition of 'IsPhysicianTo'.

In a similar way, the fact that it is permissible to inform all paramedic staff (group h) with information ϕ can be formalized as

$$\neg O(Bi(\Delta) \rightarrow ([i\ STIT\alpha\colon A^{-}])) \quad \text{where } A^{-} \stackrel{\text{def}}{=} \neg Kh\phi. \tag{3}$$

Next, there are two key decisions to make. The first one is the decision whether we model the chosen beliefset under openworld or closedworld semantics, and the second one concerns the choice which operator to take as primitive. Let us consider obligations under closedworld semantics first. Every tuple in the beliefset is true by definition. This covers all obligations, $O(\Phi)$, and everything forbidden $O(\neg\Phi)$. Every tuple that is not in the beliefset is false by definition, and thus either gratuitous, $\neg O(\Phi)$, or permissible $\neg O(\neg\Phi)$. This approach results in a rather permissive set-up because it might be very hard, even in a limited application domain, to list everything that is forbidden. As long as something is not explicitly forbidden or obligatory the agent is free to act. In our example, with the case of privacy protection, this seems too permissive.

Taking permission, $P(\Phi)$, as the primitive operator gives the exact opposite. All positive tuples are either permissible or gratuitous. Under closedworld semantics this is a restrictive form: if it is not allowed it is forbidden. This might be a very suitable option in our example. Everyone, who is allowed to be informed, is listed, and, per default, anyone who is not listed is not allowed to know. However, there might be exceptions on these rules needed. For example, in the case of an accident, paramedics may need information, which shows that this approach might be too restrictive.

In an openworld semantics the status of the proposition is undetermined. This means that a state that is not in the database might be obligatory or gratuitous, forbidden or permissible. This situation makes sense only if the agent is given

the ability to search for additional information that will allow it to classify a proposition. If the additional information is not available or inconclusive, one has to fall back to assigning the proposition a truth value on some other basis or by default. The openworld semantics is more permissive and allows for situation of greater degree of uncertainty and independent acting on the side of the agent. Taking P as the primitive operator in an openworld semantics helps taking away some edges of the restrictive nature requiring not only to list everything that is gratuitous and permissible, but also everything that is explicitly forbidden. But that might be difficult as we pointed out above.

Under openworld semantics (*ow*) the choice for the operators O or P is equivalent. It would be equivalent for closedworld semantics (*cw*) if it is possible to exhaustively list all relevant states and actions. The very reason for undertaking the effort to extend agents with moral reasoning capability, however, is the very fact that it often is not possible to do.

We further observe here that by arranging the options from least to most restrictive, we get:

$$Ocw > Oow = Pow > Pcw. \tag{4}$$

Agents have particular goals (desires) they want to achieve. Given that the agent has a view of the resulting state, it needs to evaluate the means (actions) by which these goals can be achieved in moral terms. The approach is to take these actions and run them against the knowledge base of permissible states. The tuple of permissions will contain a query. The envisioned state provides the input parameters and the query returns a boolean indicating whether the outcome or action is permitted.

Let us look at the above example. Lying, intentionally misinforming, is an undesirable trait that can be functional in achieving the agent's goal of increasing the chances of the hospital receiving the money for the treatment. How can an intention, that is a plan that the agent has chosen, be checked for its moral admissibility? We model lying as informing other agents or humans about a state Φ that the agent believes to not hold true, $\neg\Phi$. A generalized check would consist of checking all information provided to other agents or humans against the agent's knowledge base (beliefsets). The moral knowledge base contains propositions as defined above.

The next question is how the agent knows that a particular action comes under the *Inform* action-type in the beliefset. Here we have two options. If the agents are equipped with strong cognitive capabilities they can learn how to classify actions. They would learn that informing basically consists of a source, one or more targets, some information, etc. The other option is to strong type actions to allow classification. This requires more upfront design effort. Action-types would have to be determined upfront at design-time, while the actual actions and their typing can be done at runtime. The cognitive demands in this approach are considerably less. The basic moral rules would be defined upfront. As this is meta-level definition of permissible, obligatory, etc. actions or states, it allows ample room for uncertainty at design time. Strong typing would allow agents to classify their plans at run time under a moral regime.

We hope to have already demonstrated with the above analysis of moral knowledge that already relatively simple considerations lead to complex modelling questions and decisions. It will also be clear that though some of these considerations might easily apply to humans, they do not to robots and softbots. Moral philosophy and modal logic provide a powerful toolset for designers wishing to include moral considerations into their design.

3.3 Connecting Moral Knowledge to Intentions

An intention in the BDI-model is a plan that the agent has committed to. Before making the commitment, i.e. choosing a plan, the agent should evaluate the various plans that are available on the moral merits. Each plan has a *context*() method. This method contains a logical proposition with logical variables. This allows us to run the proposition against beliefsets and find bindings for the logical variables. If the plan contains information provision, the plan is typed as informing. The information to be provided is matched against the agent's beliefset. The structure of the plan and the values are matched against the permissions beliefset which will indicate whether the plan is morally permissible. The permission can be contingent on the application domain (sphere). What is permissible in one context may be impermissible in another, given the roles of the actors involved, the object of the data, etc.

In our approach, the context method contains a check against the moral knowledge database and returns the proposition we have modelled above. Together with the value of the information to be provided and a logical variable that is run against the agent's beliefsets, the proposition would evaluate the moral admissibility of the plan. If there would be no binding for the logical variable, this indicates that the agent does not hold this piece of information to be true.

The implementation of proposition (3) would contain at least three elements. First, a beliefset containing the information the agent holds to be true (or false). Second, a beliefset containing the deontic constraints. Third, a plan to do something. This plan also contains the *context*() method.

The beliefset '`MoralObligations`', listed above, is extended with a query to check the intended action against the moral constraints. It checks whether the intended plan comes under a particular action-type for which there are moral constraints, and whether the agent consider the information it wishes to communicate to be true.

2. Listing 'beliefset MoralObligations with query function'

```
public beliefset MoralObligations extends
OpenWorld {
    ...
    #indexed query getAct
    (String strType, boolean bAct);
    #complex query boolean getObligation
    (String strAct, String strType){
       boolean bAct;
       return getAct(String strAct,
       String strType, boolean bAct) &&
    getTrue(String strData, boolean bTrue);
    }
    #function query getAllObligatoryActs (){
       ...
    }
}
```

The context() method would make a reference to the above believe set and check its relationship beliefset to establish, for example, the relationship between the data object and the intended receiver of the information. Patient information can be transferred to the attending physician but not to the insurer. The context() method is a logical proposition containing queries against the beliefsets. The beliefset queries return a logical truth value. Only if the complete proposition evaluates as true will the plan be executed.

In our example we have a so-called negative duty (the obligation to refrain from doing something, i.e. from informing the insurer about private medical patient data), and a positive duty (the obligation to do something, i.e. to inform the patient's GP). The implementation strategy outlined above concerns the negative duties. The implementation of positive duties is in fact much easier. It consists of having a plan to respond to an event or state. A change in patient data does trigger an event, an observer that is attached to the data. This event in turn is handled by the agent that does have a plan to respond to the event, see for a more detailed description [24].

The challenge is designing multi-agent systems in which agents cooperate, either as parts of one larger system (also an agent in our terminology), or as ad-hoc co-operators, is to restrain the behaviour of the system at macro-level. Restricting the execution of individual plans is one thing; but if these plans call on other plans, calling in turn on other plans chosen at run-time, it becomes rather difficult. This issue has, to our knowledge, not yet been addressed at any length in a satisfactory fashion. Also in the current approach there is no more than an outline yet. Our initial approach is to have an execution of plans in a 'sand-box' where the outcomes are checked against the initial restraints defined in the originating plan before the actual execution is allowed. For the current purposes further discussion is out scope. We highlight it though as an important aspect for future research.

3.4 Meta-Level Reasoning

The last element to consider here is the situation in which goals and moral considerations are not compatible. It can be that either the agent cannot achieve its goal without breaking a moral rule, or, there are conflicting moral obligations, permissions, etc. Before discussing how to address this issue we need to describe the mechanism by which agents select the plans that potentially serve its purpose. A goal and an event are the two basic triggers to bring an agent into action. In response to the triggers all plans that are relevant are gathered using a *relevant*() method. Next, the *context*() method is executed, which further restricts the set of available plans for consideration. If there remain multiple plans, the agent has three basic mechanisms: prominence, precedence and meta-level reasoning. Prominence refers to the order in which the plans are hard wired into the agents' make-up. This constitutes some equivalent of our immediate responses. Precedence is the mechanism by which the plans are ordinally ranked. This reflects a considered hierarchy of obligations, permissions, etc. The meta-level reasoning allows the agent to gather additional information and reason about which plans to adopt. If there are conflicting moral considerations, and there is no predefined hierarchy, there are moral mechanisms to deal with this. The lying would be caught as impermissible to start with. The conflicting remaining obligations can be dealt with by using the moral algorithms at meta-level reasoning. To support the actual design of such reasoning and ranking algorithms, there is ample literature in the field of moral philosophy dealing with, for example prima facie duties, value commensurability, reflective equilibrium, etc.

If the set of plans is empty, due to the moral restrictions in the *context*() method the agent cannot act. This may, of course, seem unsatisfactory, but it might have been exactly the purpose: not allowing the available action to execute because they violate some obligations is what was intended from the start. The designer might also decide to adopt a less restrictive model of behaviour constraint and determine to reconsider the plans using the meta-level reasoning mechanism.

4 Conclusions and Future Research

We have argued that constraining behaviour of agents in multi-agent systems is a necessary feature to make them more acceptable to their human counterparts. Behaviour constraints can take many forms: permissions and obligations to do something, duties, gratuitous actions, optional actions, etc. This is a complicated domain of analysis. Behaviour constraint and direction is the core subject matter of moral philosophy. Taking moral philosophy as the starting

point of the analysis provides us with a well analysed set of concepts that defines the concepts and their relationships. Even though to aim is not to create human morality in an artificial setting, the concepts are considered as very useful.

Moral philosophy, however, is seldom defined in sufficient precise terms and relationships to be directly useful for the implementation in multi-agent systems. The connecting bridge, as it were, between moral philosophy and multi-agent systems is formed by the modal logic framework DEAL, and the BDI-model. The feasibility of this approach is demonstrated through the pseudo code in the JACK agent software.

The analysis of moral knowledge shows that there is a wide range of modes to model moral knowledge. The impact of using obligations as the primitive operator in conjunction with closedworld semantics wields a very restrictive model, in contrast to a permission based model with openworld semantics.

Our integral approach provides a complete approach from theory to software implementation of constraints on agent behaviour. It has a well-founded theoretic base, a formal expression in two models, and a demonstrated implementation facility.

As presented, our approach still contains various white-spots and drawbacks. In this paper we restricted ourselves to a particular branch of normative ethics: deontological ethics. This branch focuses on the permissibility of actions as such without reference to the outcome. In particular application settings this will be sufficient and adequate. Other applications, however, might require different considerations. This is demonstrated through the approach to side-effects. A deontological approach to norms disregards the outcomes in the evaluation of an act. Thus, side-effects do not enter into the account. In a teleological approach they feature prominently in the evaluation of an action. Deontological moral theories form a long standing tradition and it is one of the main branches of moral philosophy. It provides a good starting point for the reasoning about and modelling of moral knowledge and behaviour restraints. The complexity of our relatively simple application shows the importance of engaging moral philosophy into the design of multi-agent systems. System designers might automatically be drawn towards a particular approach without knowing the impacts of this (implicit) choice. As one of the next steps teleological analysis needs to be included as well in our framework. Without it the framework is not complete and applications are necessarily more limited. Certain applications will require an evaluation of the outcomes. This goes also for the mechanism to compare and rank conflicting obligations, duties, rights, etc. The explicit modelling of concrete mechanisms is something we have not touched upon in this paper, an omission we are well aware of and that needs to be addressed in future research. Another issue for further attention is a development of moral epistemology in artificial contexts. Strong typing actions, knowledge, etc. is currently the chosen option, but it is limiting in potential width of the application domain.

Further research will need to look into moral learning and the development of more powerful cognitive algorithms. This is certainly not a trivial exercise and still much needs to be done in order to create artificial agents with dedicated moral reasoning capabilities that can be trusted and, therefore, be used in all kinds of practical, usually dynamic situations of life. We wish to suggest here that the development of artificial moral agents within specific, well-focused domains should be tried first since in these specific domains, we expect the solution spaces to be relatively small and the complexity of moral reasoning solutions more manageable. In order to realize general acceptance of artificial agents through enhanced moral reasoning capabilities, a lot of testing is needed where agents may be used in an advisory role first, before we trust them with full decision-making power.

References

1. Agent Oriented Software Pty. Ltd. JACK (2002). www.agentsoftware.com.au
2. Artikis A, Pitt J, Sergot MJ (2002) Animated specifications of computational societies. In: Proceedings of autonomous agents and multi-agent systems (AAMAS), Bologna, pp 1053–1062
3. Bates J (1994) The role of emotion in believable agents. Commun ACM 37(7):122–125
4. Bratman ME (1987) Intention, plans and practical reasoning. Harvard University Press, Cambridge
5. Bratman ME, Israel DJ, Pollack ME (1991) Plans and resource-bounded practical reasoning. In: Pollock J, Cummins R (eds) Philosophy and AI: essays at the interface. MIT Press, Cambridge, pp 7–22
6. Boella G, van der Torre L (2004) Fulfilling or violating obligations in normative multiagent systems. In: IAT, pp 483–486
7. Danielson P (1992) Artificial morality. Routledge, London
8. Danielson P (ed) (1998) Modeling rationality, morality and evolution. Oxford University Press, New York
9. Dastani M, Hulstijn J, van der Torre L (2001) The BOID architecture: conflicts between beliefs, obligations, intentions and desires. In: Proceedings international conference on autonomous agents
10. Dastani M, Hulstijn J, Dignum MV, Meyer JC (2004) Issues in multiagent system development. In: Proceedings AMAAS
11. Dignum F, Kinny F, Sonenberg L (2002) From desires, obligations and norms to goals. Cogn Sci Q 2(3–4):407–430
12. Dumas M, Governatori G, ter Hofstede AHM, Oaks P (2002) A formal approach to negotiating agents development. Electron Commer Res Appl 1(2):193–207
13. Georgeff MP, Pell B, Pollack ME, Tambe M, Wooldridge M (1998) The belief-desire-intention model of agency. In: ATAL, pp 1–10
14. Governatori G, Rotolo A (2004) Defeasible logic: agency, intention and obligation. In: Deontic logic. Lecture notes in computer science, vol 3065. Springer, Berlin, pp 114–128

15. Halpern J (2000) On the adequacy of model logic, II. Eletron. News J. Reason. Action Change
16. Hoven MJ, van den Lokhorst GJ (2002) Deontic logic and computer supported computer ethics in cyberphilosophy, Bynum et al (eds)
17. IBM, Kasparov vs DeepBlue. The rematch, web page: http://www.research.ibm.com/deepblue/
18. Moor JH (2006) The nature, importance, and difficulty of machine ethics. IEEE Intell Syst 21(4):18–21
19. Raymond YK (2005) Lau, Adaptive negotiation agents for e-business. In: ACM proceedings of the 7th international conference on electronic commerce, Xi'an, China, pp 271–278, ISBN:1-59593-112-0, 2005
20. Sergot M, Richards F (2001) On the representation of action and agency in the theory of normative positions. Fundam Inform 48(2–3):273–293
21. Vrancken J, Van den Berg J, Dos Santos Soares M (2008) Human factors in system reliability: lessons learnt from the Maeslant storm surge barrier in the Netherlands. J Crit Infrastruct 4(4):418–429
22. Walzer M (1983) Spheres of justice. Basic Books, New York
23. Wiegel V, Van den Hoven MJ, Lokhorst G-J (2005) Privacy, deontic epistemic action logic and software agents, an executable approach to modeling moral constraints in complex informational relationships. Ethics Inf Technol 7(4):251–264. doi:10.1007/s10676-006-0011-5
24. Wiegel V (2007) SophoLab. A laboratory for experimental philosophy, Delft
25. Wooldridge M (2000) Reasoning about rational agents. MIT Press, Cambridge
26. Wooldridge M (2002) MulitAgents systems. Wiley, Chichester
27. http://en.wikipedia.org/wiki/Intelligent_agent

Vincent Wiegel is a researcher at the Delft University of Technology and the 3TU centre for Ethics and Technology. In a joined research project with KPN, TNO and TU Delft he looks into the question how deep (moral) values are to be incorporated in the innovation process. This process stretches from requirement engineering, modelling and design, to implementation. His main associate in this research is Luuk Simons. Research questions that are tackled include 'What does it take to create an artificial agent, a robot, a computer(network) that is capable of performing some form of moral reasoning?' and 'What level of moral reasoning is attainable given our current understanding of morality and the current technology available?' Using a modal logic framework DETAIL (developed by Gert-Jan Lokhorst en Jeroen van den Hoven) and Bratman's BDI model and multi-agent software systems, artificial agents are created that display some basic form of moral behaviour. These components are joined in 'SophoLab', a laboratory for philosophical experimentation, set-up by Vincent. Vincent also developed a methodology for philosophical experiments.

Jan van den Berg is currently Associate Professor within the section of ICT at the Faculty of Technology, Policy and Management of Delft University of Technology, The Netherlands. He holds an engineering diploma in Applied Mathematics from the same university. His PhD was in the area of Artificial Intelligence and devoted to a mathematical-physical analysis of Recurrent Neural Networks. He published, with many different people, in many different international journals and conference proceedings on subjects related to ICT for improving business, government and other organizations. These publications concern (application) domains like Computational Intelligence (including (Probabilistic) Fuzzy Systems, Computational Finance, Business Intelligence, Intelligent Knowledge Discovery, and Computational Philosophy), Information Security and Privacy Enhancing Technologies, Sustainable Development (especially in the domain of Agriculture), and Healthcare.